

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-208981

(43)Date of publication of application : 26.07.2002

(51)Int.Cl.

H04L 29/06

(21)Application number : 2001-004399 (71)Applicant : HITACHI LTD

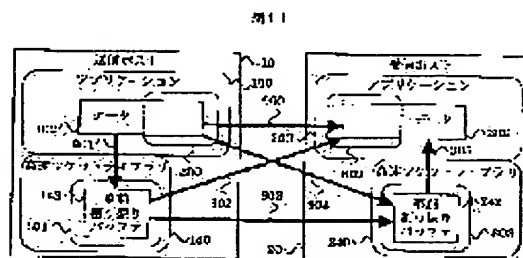
(22)Date of filing : 12.01.2001 (72)Inventor : MASHIERU FUREDERIKO

(54) COMMUNICATION METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To obtain a high speed communication method using socket API or MPI API.

SOLUTION: Five novel functions are used. (1) The receiving side informs the transmitting side of a data length for determining which of reception at application data 202 or reception at a previously assigned buffer 242 is optimal. (2) An effect of informing the reception address of the application data 202 is calculated and information is suppressed if the effect is low. (3) A communication protocol enabling eight communication methods is used. (4) A transfer data length expected for transmitting/receiving operation is informed previously to the opposite party. (5) Previously assigned buffers 142 and 242 are altered (extension, contraction, addition, deletion, and the like) according to a communication pattern. According to these functions, high speed communication is attained while reducing the overhead of processing and the amount of memory being used.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-208981

(P2002-208981A)

(43) 公開日 平成14年7月26日 (2002.7.26)

(51) Int.Cl.⁷

H 0 4 L 29/06

識別記号

F I

H 0 4 L 13/00

テーマコード(参考)

3 0 5 C 5 K 0 3 4

審査請求 未請求 請求項の数17 O L (全 11 頁)

(21) 出願番号 特願2001-4399(P2001-4399)

(22) 出願日 平成13年1月12日 (2001.1.12)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 マシエル フレデリコ

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(74) 代理人 100075096

弁理士 作田 康夫

Fターム(参考) 5K034 AA01 AA05 HH63 MM39

(54) 【発明の名称】 通信方法

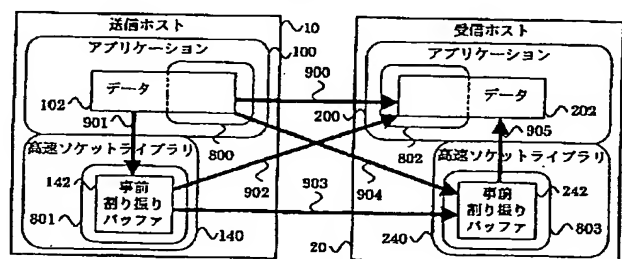
(57) 【要約】

【課題】 ソケットAPIやMPI APIを使用した通信の高速化

【解決手段】 5つの新機能を使用する。(1) 受信側が、アプリケーション・データ202での受信と事前割り振りバッファ242での受信のどれが最適かを決定するデータ長を送信側知らせる。(2) アプリケーション・データ202の受信アドレスを知らせる効果を計算し、効果が低い場合に知らせを抑える。(3) 8つの通信方法を可能にする通信プロトコルを使用する。(4) 送受信動作に期待される転送データ長を通信相手にあらかじめ知らせる。(5) 通信パターンにより事前割り振りバッファ142, 242を変更する(拡大・縮小・追加・削除等)。

【効果】 通信を高速化し、処理オーバーヘッドとメモリ使用量を減らす。

図11



【特許請求の範囲】

【請求項 1】通信手段を介して情報処理装置間でデータを転送する通信方法であり、受信側となるべき第一の情報処理装置は送信側となるべき第二の情報処理装置に対し、前記データの受信対象のメモリ領域を連絡するようにされた通信方法において、前記第一の情報処理装置は前記第二の情報処理装置に対し、前記受信対象のメモリ領域を第二の情報処理装置から指示して該受信対象のメモリ領域に前記データを転送する第一の転送動作と、前記第一の情報処理装置に予め割り振ったバッファ領域を介して該データを転送する第二の転送動作との何れを選択すべきかを判定するための転送データ長に関する閾値を通知することを特徴とする通信方法。

【請求項 2】前記閾値は転送のスループットを向上するために定められることを特徴とする請求項 1 の通信方法。

【請求項 3】前記閾値は転送のレイテンシーを削減するために定められることを特徴とする請求項 1 の通信方法。

【請求項 4】前記閾値は転送の処理量を削減するために定められることを特徴とする請求項 1 の通信方法。

【請求項 5】通信手段を介して情報処理装置間でデータを転送する通信方法であり、受信側となるべき第一の情報処理装置は送信側となるべき第二の情報処理装置に対し、前記データの受信対象のメモリ領域を連絡するようにされた通信方法において、前記第一の情報処理装置は前記第二の情報処理装置に対し、前記受信対象のメモリ領域を第二の情報処理装置から指示して該受信対象のメモリ領域に前記データを転送する第一の転送動作と、前記第一の情報処理装置に予め割り振ったバッファ領域を介して該データを転送する第二の転送動作との何れを選択すべきかを判定するための転送データ長に関する閾値を通知し、前記第二の情報処理装置は転送すべきデータ長が前記閾値を越えるか否かにより前記第一の転送動作か、第二の転送動作かを決定して前記データを転送することを特徴とする通信方法。

【請求項 6】通信手段に接続し、上記の通信手段を介して第二情報処理装置からデータを受信し、上記の通信手段でデータを受信する前に対象のメモリ領域を受信可能な領域として指示する第一の情報処理装置において、上記受信可能な領域として指示する動作の処理時間により、あらかじめ割り振って指示したメモリ領域の大きさを決定し、上記のメモリ領域の大きさを上記第二情報処理装置に知らせ、第二情報処理装置に上記メモリ領域の大きさを超えないデータ長の送信を上記あらかじめ割り振って指示したメモリ領域に送信してもらい、超えるデータ長の送信に対象のメモリ領域を指示して上記対象のメモリ領域に送信してもらうことにより、最速の通信方法を使用することを特徴とする通信方法。

【請求項 7】通信手段を介して情報処理装置間でデータ

を転送する通信方法であり、受信側となるべき第一の情報処理装置は前記データの受信対象のメモリ領域を登録し、前記受信対象のメモリ領域のアドレスを送信側となるべき第二の情報処理装置に対して通知することを特徴とする通信方法。

【請求項 8】前記第一の情報処理装置は、前記受信対象のメモリ領域の登録が必要か否かを判定し、必要があった時にのみ前記メモリ領域の登録と前記アドレスの第二の情報処理装置に対する通知とを実行することを特徴とする請求項 7 の通信方法。

【請求項 9】前記判定は前記アドレスの通知の効率を測定することにより実行することを特徴とする請求項 8 の通信方法。

【請求項 10】通信手段に接続し、上記の通信手段を介して第二情報処理装置にデータを送信し、上記の通信手段でデータを送信する前に対象のメモリ領域を送信可能な領域として指示する第一の情報処理装置において、あらかじめ割り振って指示したメモリ領域に送信データをコピーし、上記コピーしたデータのアドレスとデータ量を上記第二情報処理装置に知らせ、上記第二情報処理装置にデータを読み出すことを特徴とする通信方法。

【請求項 11】通信手段に接続し、上記の通信手段を介して第二情報処理装置にデータを送信し、上記の通信手段でデータを送信する前に対象のメモリ領域を送信可能な領域として指示する第一の情報処理装置において、あらかじめ割り振って指示したメモリ領域に送信データをコピーし、上記コピーしたデータを、上記第二情報処理装置がこの通信に指示したメモリ領域に送信することを特徴とする通信方法。

【請求項 12】通信相手のメモリアドレスを指定しデータを送信できる通信手段に接続し、上記の通信手段を介して第二情報処理装置からデータを受信する第一の情報処理装置において、第二情報処理装置がこのデータ転送に指示しアドレスとデータ量を知らせたメモリ領域から、第一情報処理装置があらかじめ割り振って指示したメモリ領域に読み出すことを特徴とする通信方法。

【請求項 13】通信手段に接続し、上記の通信手段を介して複数のデータ転送方法を持つ通信プロトコルで送受信する第一と第二の情報処理装置において、送受信開始時、第一およびまたは第二の情報処理装置が通信相手に平均転送データ長を知らせ、上記平均転送データ長により転送方法を選択することを特徴とする通信方法。

【請求項 14】上記転送方法の選択は、対象のメモリ領域を指示して送受信するか否か、およびまたはあらかじめ割り振って指示したメモリ領域を介してデータを送受信するか否かを特徴とする請求項 13 の通信方法。

【請求項 15】通信相手のメモリアドレスを指定しデータを送受信できる通信相手のメモリアドレスを指定しデータを送信できる通信手段に接続し、上記の通信手段を介して第二情報処理装置とデータを送受信し、上記の通

信手段でデータを受信する前に対象のメモリ領域を受信可能な領域として指示し、あらかじめ割り振って指示したメモリ領域を介してデータを送受信する第一の情報処理装置において、上記あらかじめ割り振って指示したメモリ領域を変更することを特徴とする通信方法。

【請求項16】上記変更が上記メモリ領域の拡大およびまたは縮小であることを特徴とする請求項15の通信方法。

【請求項17】上記あらかじめ割り振って指示したメモリ領域は受信用途と受信用途に分かれおり、上記変更が受信用途のメモリ領域を送信用途にすること、およびまたは送信用途のメモリ領域を受信用途にすることを特徴とする請求項15の通信方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、複数の種類の通信網により接続された複数の計算機を有する計算機システムにおける、計算機間のデータ送受信方法に係り、特に計算機間メモリ間データ転送の機能を持つネットワークとハードウェアの上での計算機間データ送受信方法に関する。

【0002】

【従来の技術】計算機間通信、特にインターネットやイントラネットでの通信には、TCP/IPプロトコルが極めて一般的に使用されている。TCP/IP処理をアプリケーションでなくオペレーティングシステムが行うため、アプリケーションがTCP/IPで通信するために「ソケット」と呼ばれるAPI (Application Programming Interface、アプリケーションがコンピュータやオペレーティングシステムのある機能を用いるために呼び出す関数の集合)を用いる(W. Richard Stevens, "UNIX Network Programming," Prentice Hall, U.S.A., 1990, ISBN 0-13-949876-1参照)。

【0003】図1にTCP/IPプロトコルを使用し通信するホストのソフトウェア構成例を示す。ホスト10はネットワーク18を使用して通信する。ホスト10のオペレーティングシステムのカーネル120がTCP/IPのプロトコル処理121をし、通信ハードウェア11を制御して通信する。アプリケーション100のプログラム101がソケットAPI90を用い、ライブラリ110を呼び出す。ライブラリがシステムコール111を実行してカーネル120を呼び出す。カーネル120がソケット用バッファ122を介して、アプリケーション100のデータ102を送受信する。

【0004】TCP/IP通信はプロトコル処理121の処理量が多く、そしてシステムコール111と、データ102とソケットバッファ122の間のコピーはオーバーヘッドとなるため、これらの処理は通信性能を制限することがある。このため、スーパーコンピュータやワークステーションクラスのような、高速な通信を必要

とする計算機システムでは、プロトコル処理、システムコールとデータコピーをせず、カーネルを介さずにアプリケーション間データ転送ができるネットワークが用いられる。本明細書では今後、この通信方法を「高速通信」と呼ぶ。高速通信の例としてVIA (Compaq Computer Corp.; Intel Corp., Microsoft Corp., "Virtual Interface Architecture Specification, Draft Revision 1.0," December 4, 1997, <http://www.viarch.org>参照)がある。高速通信とTCP/IPは機能が異なるため、これらAPIも異なる。

【0005】図2に高速通信を使うホストのソフトウェア構成例を示す。アプリケーション103のプログラム104が高速通信API91を用いて、高速通信ライブラリ130を呼び出し、データ105を送受信する。高速通信ライブラリ130の通信処理131はカーネル120を介さずに高速通信ハードウェア12を起動しデータ105を高速通信ネットワーク19で通信する。高速通信におけるデータ送受信では、アプリケーション103が送受信したいデータ105のアクセス権限があるかどうかという検査、そしてアプリケーション103が指定した仮想アドレスを高速通信ハードウェア12が使う物理アドレスへの変換という二つの処理が必要である。このためアプリケーション103が送受信する前に、高速通信ライブラリ130を呼び出し、送受信するデータ105を登録する(登録されたデータを807のような角丸四角形で示す)。登録処理を高速通信ライブラリの呼び出し(132)でカーネルが行う(123)ため、アクセス権限を調査し、権限があった場合にアドレス変換を行い、登録したデータをメモリ登録テーブル13に登録することができる。高速通信ハードウェア12がこのメモリ登録テーブル13を用い、アクセス権限調査とアドレス変換を行う。高速通信API91はソケットAPI90と異なるため、ソケットAPI90を使うアプリケーション100が高速通信を使用するためには、アプリケーション100を高速通信API91に向けて書き換えなければならない。この書き換えは難しいため、多くのアプリケーションが変更されず従来のソケットAPIを使い続け、高速通信の高速性を活用できない。この問題を解決するために、図3に示す「高速ソケット」という方式を用いる。高速ソケットライブラリ140はアプリケーション100のソケットAPI90の呼び出しを受け、エミュレーション処理141をし、高速通信を用い通信する。このため、アプリケーションの互換性を保ちながら、高速通信の高速性を用いることができる。高速ソケットの例として、公開特許公報特開平11-328134、Berkeley大学の方式(S. H. Rodrigues, T. E. Anderson, D. E. Culler, "High-Performance Local Area Communication With Fast Sockets," Proceedings of the USENIX '97, 1997, pp. 257-274参照)、Shahらによる方式(H. V. Shah, C. Pu, R. S. Madukkarumu

kumana, "High Performance Sockets and RPC over Virtual Interface (VI) Architecture", Proceedings of CANPC'99, 1999参照)、Microsoft社のWinsock Direct ("Winsock Direct Specification", Microsoft Windows Driver Development Kit (DDK) 参照) が挙げられる。

【0006】アプリケーション100のデータ102を登録(800)して通信した場合、バッファ登録800の処理オーバーヘッド(132, 123)が生じる。データ長が長い場合にこのオーバーヘッド(132, 123)は通信時間に比べて短いため、高速性を得られる。一方、データ長が短いとき、通信時間に比較してこのオーバーヘッドは大きく、通信性能が低下する。この問題を解決するため高速通信ライブラリ140は起動時に、事前割り振りバッファ142をアロケートし登録(801)する。短いデータ102を通信するとき、このデータ102を登録せず事前割り振りバッファ142にコピーし通信する。この場合にはコピーのオーバーヘッドが生じるが、データ長が短くこのオーバーヘッドが登録処理に比較して少ないため、高速性を得られる。事前割り振りバッファ142は普段送信用バッファと受信用バッファに分かれているが、図3と今後のソフトウェア構成の図ではこれらをまとめて一つのバッファ142として示す。

【0007】以上はTCP/IP通信と高速ソケットの説明であった。一般アプリケーションがTCP/IP通信(と、その結果、ソケットAPI)を用いる一方、科学技術計算アプリケーションはMPI(Message Passing Interface Forum, "MPI: A Message-Passing Interface Standard," 1995参照)のようなAPIを用いる。MPIは計算機アーキテクチャ非依存のため、高速通信の上でMPIをインプリメントする場合、MPIのAPIの呼び出しを高速通信のAPIの呼び出しにマッピングする。この機能を実現する製品としてMPI Software Technology社のMPI-Proが挙げられる(R. Dimitrov and A. Skjellum., "Efficient MPI for Virtual Interface (VI) Architecture," Proceedings of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, Nevada, U.S.A., June 1999, Vol.6, pp. 3094-3100参照)。図4にMPIの実現方法を示す。図4では、MPIを使用するアプリケーション106のプログラム107がMPI API 92を利用してデータ108を通信する。MPIライブラリ150がエミュレーション151を行い、上記のマッピングを行う。MPI(図4)の構成は高速ソケット(図3)の構成と同様のため、両者の通信における課題も同様である。本明細書では記載がなければ、高速ソケットに説明する方法はMPIにも当てはまり、またMPIに説明する方法は高速ソケットにも当てはまる。

【0008】

【発明が解決しようとする課題】本発明は従来の高速ソケットライブラリ140やMPIライブラリ150のような通信ライブラリの5つの問題を解決する(下記にこれらのライブラリを「エミュレーションライブラリ」と呼ぶ)。ここではこれらの問題を概説して、必要な場合に発明の実施の形態の説明ではこれらの問題の詳細な説明をしてから本発明の解決手段を説明する。第一の問題は次のとおりである。従来方式では送信ホストがデータ長により、送信ホストにデータ102, 108を登録(800, 808)した通信と事前割り振りバッファ142, 152にコピーした通信のどちらが最適かを選択するが、受信ホストにどちらが最適かを考慮しない。このため、受信ホストの受信処理性能を低下する。

【0009】第二の問題は次のとおりである。受信ホストで受信呼び出しが受信データよりを先行した場合、受信ホストが受信データ102, 108領域を登録(800)してこのアドレスとデータ長を通信相手に知らせることができる。しかし、送信ホストが送信開始後にこの知らせを受信した場合、この知らせは無駄となり、送信ホストと受信ホストの処理オーバーヘッドとなり、ネットワークバンド幅を占めるため、システム全体の処理性能を低下する。

【0010】第三の問題は次のとおりである。従来方式は送信ホストからのデータ書き込みと受信ホストからのデータ読み出しという二つのデータ転送方法と、受信ホストと送信ホストそれぞれのデータ102, 108を登録(800, 808)した通信と事前割り振りバッファ142, 152にコピーした通信の4つの組み合わせ、全体で8つの組み合わせを全て利用することができない。このため、高速通信を可能にするネットワークの性能を最大限に向上できない。

【0011】第四の問題は次のとおりである。従来方式は通信相手にもかわらず同じ通信方法を使用する。しかし、今後は通信相手がサーバ等のコンピュータでなく、iSCSI(TCP/IP上SCSIプロトコル、J. Satran et al., "iSCSI (Internet SCSI)," Internet Engineering Task Force Internet-Draft draft-satran-iscsi-01.txt, July 10, 2000参照)を使用しているストレージ装置であることが考えられる(本発明では、通信する装置を種類にかかわらず「ホスト」と呼ぶ)。ストレージ装置はコンピュータに比較して事前割り振りバッファ142に使用できるメモリ量が制限されており処理性能が低いことがあるため、上記第三の問題に述べた8つの組み合わせの一部のみが効率的である。通信相手の特性により通信方法を制限しないことは、例えばこの通信相手がストレージ装置の場合には装置の必要となるメモリなどを増加し、送受信処理を複雑にし装置の必要な処理能力を高め、コストを高くする。

【0012】第五の問題は次のとおりである。従来方式

はTCP/IP接続確立時に事前割り振りバッファ142, 152をアロケートし、この後の通信にはバッファの大きさ等を変更しない。このため、このTCP/IP接続の特性に必要なバッファ量を適応することができない。例えば必要な時にバッファの大きさを増加しないことは性能を低下する要因になる。そして、事前割り振りバッファ142, 152のような登録(801, 809)したデータ領域は、データ送受信対象のためスワップアウトできなく、主記憶を占める。このため、バッファの大きさを削減しないことは他のアプリケーションが使えるメモリを少なくするため性能低下の要因にもなる。

【0013】

【課題を解決するための手段】第一の問題の解決方法は、通信するホストが通信相手にデータ102, 108を登録(800, 808)した通信と事前割り振りバッファ142, 152にコピーした通信のどれが最適かを決定するデータ長を知らせることである。

【0014】第二の問題の解決方法は、受信ホストが知らせの効果を計算し、効果が低い場合に知らせを抑えることである。

【0015】第三の問題の解決方法は、8つの組み合わせを可能にする通信プロトコルである。

【0016】第四の問題の解決方法は、送受信動作に期待される転送データ長を通信相手に知らせることである。

【0017】第五の問題の解決方法は通信パターンによるバッファの変更である。

【0018】

【発明の実施の形態】<<第一の問題の解決方法>>この問題の解決方法の説明としてまず、従来方式を説明する。図5にMPI-Proの通信方法を示す。(今後、通信方法の図を理解しやすくするために、図3と図4のアプリケーション100, 106とエミュレーションライブラリ140, 150のみを示す。両ホスト10, 20は同様なソフトウェア構成を持つ。そして、片方向のデータ転送のみを示し、左のホストを送信ホスト10、右のホストを受信ホスト20とする。)MPI-Proは送信側では事前割り振りバッファを利用しなく、アプリケーション106のデータ108から直接送信する。全ての通信は送信ホスト10からの書き込みである。データ長が長い場合にデータ108を直接アプリケーション206データ208に送信(900)し、データ長が短い場合データを受信ホスト20の事前割り振りバッファ252に送信(902)する。ここでは、どちらに送信するか決定するホストは送信ホスト10である。

【0019】スーパーコンピュータの場合、ホスト10, 20は普段全て同じ物であるため、送信ホスト10は受信ホスト20のアプリケーションデータ208と事前割り振りバッファ252とのどちらに送信すれば最適

かを判断できる。しかし、高速ソケット通信やMPIを実行するワークステーションクラスタのようにホスト10, 20が異なるシステムの場合、ホストによりメモリ登録動作(132, 123)の時間とメモリコピーの性能が異なるため、送信ホスト10だけの判断は不可能である。判断を間違えば受信処理(と、その結果、送信ホスト10と受信ホスト20を含むシステム全体)の性能が低下する。

【0020】以上は従来技術である。本発明ではこの問題を解決するために、受信ホストが登録(805)した通信と事前割り振りバッファ252を介した通信のどれが最適かを決定するデータ長を送信ホストに知らせる。知らせるタイミングはまず、高速ソケットでは通信するホスト10, 20がソケットAPI90でソケットの接続を確立したとき、MPIではMPIライブラリ150, 250の初期化時である(今後、このタイミングを「通信開始」と呼ぶ)。従来(図6a)このタイミングで送信するデータ910(事前割り振りバッファアドレスとデータ長等)と一緒に、本発明のデータ長の知らせ911(図6b)を転送することが考えられる。そしてもう一つの可能なタイミングとして、ホスト20が始めてホスト10に通信したとき、この情報を追加することも考えられる。

【0021】どちらの通信方法が最適かを決定するデータ長の設定として、(1)アプリケーション206からの設定、(2)ホスト10, 20の管理者やユーザやアプリケーションからの設定、(3)エミュレーションライブラリ140, 150をホスト10, 20にインストールしたプログラムの設定、などの方法が考えられる(しかし、これらの方法に限られていない)。

【0022】以上の発明のため、受信ホスト20の受信処理(と、その結果、システム全体)の性能が向上する、という効果を得る。

【0023】<<第二の問題の解決方法>>この問題の解決方法の説明としてまず、従来方式を説明する。図7に従来方式を示す。受信ホスト20のアプリケーション206が受信呼び出しを実行し、エミュレーションライブラリ250が、アプリケーションデータ208に直接受信することが効率的であることを判断したとき、データ208を登録(805)して、送信側に受信アドレスとデータ長を知らせること(950)ができる(データ転送以外、エミュレーションライブラリ140, 150, 250は制御メッセージを交換し、このアドレスとデータ長の知らせを制御メッセージとして転送する)。この場合、送信ホスト10が送信呼び出しを実行したときにデータをこのアドレスに送信して(951)、そして送信完了の確認メッセージ952を送信する。このため、送信呼び出しの直後に送信の開始ができる。しかし、以前述べたとおり、送信ホスト10が送信開始後にアドレスの知らせ950を受信した場合、この知らせ9

50は無駄となり、処理オーバーヘッドとなり、ネットワークバンド幅を占めるため、システム全体の処理性能を低下する。

【0024】以上は従来の技術である。本発明はこの問題を解決するために、受信ホスト20がアドレスの知らせ950の効果を計算し、効果が低い場合に知らせを抑える。送信したアドレスの知らせ950の送信回数に対して、このアドレスに受信した回数の割合で効果を計算できる。そして、この効果があるしきい値より低い場合、アドレスの知らせ950の送信を抑える。

【0025】上記の解決方法にはまず、ユーザや管理者、エミュレーションライブラリ140、150、250作者がインストールプログラム、あるいはアプリケーション200がしきい値を設定することが考えられる。そして、全てのアドレスの知らせ950をまとめて効果を計算すること、そして受信アドレス毎に計算すること、の2つの方式が考えられる（後者の場合、効率の悪い受信アドレスだけに、アドレスの知らせ950を抑えることができる）。そして、抑える動作として中止（止めて続けない）と中断（止めた後に続く）が考えられる。

【0026】以上の発明のため、送信ホスト10と受信ホスト20の処理効率を向上し、ネットワークバンド幅を無駄に占めないため、これらのホスト（と、その結果、システム全体）の性能が向上する、という効果を得る。

【0027】＜＜第三の問題の解決方法＞＞ここではまず、従来方式の通信方法を説明する。今後送信箇所と受信箇所の組み合わせを示す番号（900、904等）に、送信ホスト10からの書き込み（write）か受信ホスト20からの読み出し（read）を加えて各組み合わせを示す。例えば、以前説明した図5のMPI-Proは900-writeと904-writeの2つの組み合わせのみを使用する。

【0028】図8にWinsock Directの通信方法を示し、図9にプロトコルの詳細を示す。Winsock Directではまず、送信ホスト10がデータを事前割り振りバッファ142、242の間でデータ送信する（940、930）（903-write）。受信ホスト20が受信したデータをアプリケーション200のデータ202にコピーする（905、931、942）。データ長が長い場合、上記で先頭データのみを送信し、残りのデータ102を登録し（800）、その先頭アドレスを上記の送信940、930に加える。受信ホストがデータ202を登録（802）する。高速通信ハードウェア12が受信ホスト20からの読み出し通信の機能がある場合、受信ホスト20が通信データを読み出す（932、900-read）。一方、受信側からの読み出し通信機能がない場合受信ホストが受信領域の先頭アドレスを知らせ（941）、送信ホスト10がデータを書き込む（94

3、900-write）。この後、最後に通信をしたホストが通信完了の確認を送信する（933、944）。そして、両ホスト10、20がメモリ登録（800、802）を解除する。

【0029】図10にShahらによる方式の通信方法を示す。送信ホスト10はデータ長が短い場合、事前割り振りバッファ142、242間でデータを送信する（903-write）。一方データ長が長い場合データ102を登録（800）して、受信ホストの事前割り振りバッファ242に送信する（904-write）。

【0030】以上は従来方式である。本発明は、図11に示すとおり、8つの組み合わせを全て利用可能にするプロトコルを使用する。特にこのプロトコルは従来方式が利用しなかった902-read、902-write、903-read、904-readを可能にする。

【0031】以下に、本発明の通信方法を説明する。図12に送信ホスト10側のアルゴリズムを示す。まず、受信したアドレス知らせメッセージがあれば、これらのメッセージを処理する（701）。そして送信データ102、108のデータ長を調べ（702）、データが長い場合にメモリを登録（800、808）し（704）、短い場合に事前割り振りバッファ142、152にコピーする（703）。

【0032】次に、アドレス知らせメッセージで知らせた、受信ホスト20での宛先アドレスがあれば（705）、送信データを受信ホスト20のアプリケーションデータ202、208に書き込み送信する（706）

（長いデータ長の場合900-write、短いデータ長の場合902-writeになる）。宛先アドレスがなければ、受信ホスト20の事前割り振りバッファ242、252への送信が可能か（すなわち、事前割り振りバッファに空きがあるか）、そして適切か（第一の問題で説明したとおり、受信ホスト20がこのデータ長を事前割り振りバッファ42、252で受信したいか）を調べる（707）。この二つの条件が真であれば、送信ホスト10が事前割り振りバッファ242、252に書き込み送信する（708）（長いデータ長の場合904-write、短いデータ長の場合903-writeになる）。一方、この二つの条件のどちらかが真でなければ、送信データのアドレス知らせを送信して（709）、受信完了メッセージを待つ（710）（長いデータ長の場合900-readか904-readのどちらか、短いデータ長の場合902-readか903-readのどちらかになる）。最後に、送信データを解放（711）する（長いデータ長の場合登録800、808を、短いデータ長の場合事前割り振りバッファ142、152を解放する）。

【0033】図13に受信側のアルゴリズムを示す。まず、事前割り振りバッファ242、252で受信したデータをコピー（905）して、アドレス知らせメッセー

ジがあるかを調べる(721)。アドレス知らせメッセージがあった場合(722)、データ長を調べる(723)。データ長が長い場合、アプリケーションデータ202, 208を登録(802, 805)し(724)、送信ホスト10からデータを読み出す(725)(900-readか902-readのどれかになる)。一方、データ長が短い場合、受信ホスト20が事前割り振りバッファ242, 252にデータを読み出す(726)(903-readか904-readのどれかになる)。データ長にもかかわらず、最後に受信完了メッセージを送信する(727)。

【0034】アドレス知らせメッセージがなかった場合(722)、データ長を調べる(728)。データ長が短い場合、事前割り振りバッファ242, 252でのデータ受信(903-writeか904-write)か、アドレス知らせメッセージを待つ(後者の場合、図13の処理をスタート720から繰り返す)。一方、データ長が長い場合にはアプリケーションのデータを登録して(729)、この先頭アドレスをアドレス知らせメッセージで送信する(730)。送信ホスト10では送信処理開始の前にこのアドレス知らせメッセージが受信されたら、900-writeと902-writeのどれかの通信になる。一方、受信ホスト20がこのステップでアドレス知らせメッセージを受信すれば、これは送信ホスト10と受信ホスト20が同時にお互いにアドレス知らせメッセージを送信したことが分かる。この場合、送信ホスト10に送信してもらうために、受信ホスト20がこのデータ転送におけるアドレス知らせメッセージを無視する。

【0035】以上の発明のため、送信ホスト10と受信ホスト20の間の通信性能が向上し、これらのホスト(と、その結果、システム全体)の性能が向上する、という効果を得る。

【0036】<<第四の問題の解決方法>>ストレージ装置などのホスト10, 20はアプリケーションデータ102, 202, 108, 208か通信割り振りバッファ142, 152, 242, 252のどれかしか装備しないことが考えられる。第三の問題の解決方法で説明した通信アルゴリズムはこの場合にでも使用できる。あるホスト10, 20にアプリケーションデータ102, 108, 202, 208がない場合、このホスト10, 20の処理の判断702, 723, 728をいつも「短い」とする。逆にあるホスト10, 20に事前割り振りバッファ142, 242, 152, 252がなければ、このホストでこれらの判断をいつも「長い」とし、そして通信開始にこのホストから図6aの事前割り振りバッファアドレスを送信しなく、そして通信相手に判断707の「可能かつ適切か」の条件に「存在するか」という条件を加える。このため、必要でない機能のインプリメントが不要となり、そして事前割り振りバッファ14

2, 242, 152, 252がない場合このメモリ領域のアロケーションが不要となり、このアルゴリズムは容易なインプリメントと資源の節約を可能にする。しかし、下記に説明する問題が生じる。

【0037】上記のアルゴリズムを使用しホストとストレージ装置が通信している場合、ストレージ装置は必要でない資源(事前割り振りバッファ142, 242, 152, 252等)をアロケートしない。一方、ホスト側は通信の特性を理解しないため、例えばデータ転送単位がいつも長い時にでも事前割り振りバッファ142, 242, 152, 252をアロケートし、メモリを無駄にする。

【0038】本発明では上記の問題を解決するために通信初期化時に期待される転送データ長を使用してライブラリの初期化を行う。この転送データ長を通信相手に知らせ、およびまたはアプリケーション100, 200, 106, 206が指定する。この転送データ長が「長い」か「短い」により、アプリケーションのデータ送受信が必要か、または事前割り振りバッファ142, 242, 152, 252が必要かを判断できる。

【0039】以上の発明のため、ホスト10, 20の間の通信性能が向上し、メモリを節約するため、これらのホスト(と、その結果、システム全体)の性能が向上する、という効果を得る。そしてホスト10, 20に必要な処理性能とメモリ量だけを装備すればよい、システムのコストを低下できる、という効果もある。

【0040】<<第五の問題の解決方法>>次に本発明の解決方法を説明する。まず、事前割り振りバッファの変更は(1)拡大か縮小のサイズ変更、(2)追加か削除、(3)受信用バッファを送信用にすることか、送信用バッファを受信用にすること、の3種類がある。

【0041】ホスト10, 20は次の動作で変更を決定することが考えられる。まず、エミュレーションライブラリ140, 150, 240, 250の起動時に、サイズの最大値と最小値、そして使用率の上限と下限の値を設定する。これらの値の設定方法は(1)ライブラリ140, 150作成時の定数(2)ホスト10, 20のユーザや管理者やユーザやアプリケーションからの設定、(3)ライブラリ140, 150, 240, 250をホスト10, 20にインストールしたプログラムの設定、などの方法が考えられる(しかし、これらの方法に限られていない)。そして、通信開始後、送受信動作毎およびまたは定期的に送信用事前割り振りバッファ142, 152と受信用事前割り振りバッファ242, 252の使用率を調べ、平均使用率を計算する。この平均使用率が上限を超え、そしてこの事前割り振りバッファ142, 242, 152, 252のサイズが最大限を超えていない場合、バッファの拡大や追加を行う。逆に、この平均使用率が下限を超え、そしてこの事前割り振りバッファ142, 242, 152, 252のサイズが最小限

を超えていない場合、バッファの縮小や削除を行う。そして送信用バッファにある変更、そして受信用バッファにその逆の変更を決定したら、バッファの用途を変更する（逆もまた同様である）。例えば、送信用事前割り振りバッファ 142, 152 を拡大して受信用事前割り振りバッファ 242, 252 を縮小する場合、受信用バッファの一部を送信用にすることが考えられる。

【0042】受信ホスト 20 での事前割り振りバッファ 242, 252 を変更した場合、受信ホスト 20 が送信ホスト 10 に変更内容を制御メッセージで知らせる必要がある（逆に、送信ホスト 10 の送信用事前割り振りバッファ 142, 152 の変更を受信ホスト 20 に知らせる必要はない）。サイズ縮小、バッファ削除と用途変更の変更知らせメッセージの場合、送信ホストが変更される領域にデータを送信しないために、受信ホスト 20 が変更知らせメッセージを送信して、送信ホストが応答した後に変更を行う。これら以外の変更を、知らせメッセージを行う前にでも変更が行えられ、そして送信ホストの応答が不要である。

【0043】以上の発明のため、ホスト 10, 20 の間の通信性能が向上し、メモリを節約するため、これらのホスト（と、その結果、システム全体）の性能が向上する、という効果を得る。そしてホスト 10, 20 に必要なメモリ量だけを装備すればよいと、システムのコストを低下できる、という効果もある。

【0044】＜＜変形例＞＞本発明はすでに記載した実施の形態あるいはその変形例に限定されるのではなく、以下に例示する変形例あるいは他の変形例によっても実現可能であることは言うまでもない。また、上記複数の実施の形態あるいはその変形例として記載の技術あるいは以下の変形例の組み合わせによっても実現できる。

(1) 以上の説明ではデータ 102, 202, 108, 208 を登録 (800, 802, 805, 806) して通信した場合、通信完了後に登録を解除すると述べている。しかし、MPI-Pro と同様に、後で同じアドレスのデータが通信された場合に登録を不要にするために登録を解除しなくキャッシングすることが考えられる。

(2) 以上のアルゴリズムやプロトコルの説明では通信完了確認メッセージの送信を示したが、高速通信ハードウェア 12 や通信プロトコルの機能によりこれらのメッセージ、あるいはその一部が不要となる。

(3) 上記の 5 つの問題の解決方法を別々に使用すること、あるいは複数同時に組み合わせて使用することがで

きる。

【0045】なお、本発明を実施するためのプログラムは、それ単独であるいは他のプログラムと組み合わせ、ディスク記憶装置等のプログラム記憶媒体に記憶された販売することができる。また、本発明を実施するためのプログラムは、すでに使用されている通信を行うプログラムに追加される形式のプログラムでもよく、あるいはその通信用のプログラムの一部を置換する形式のプログラムでもよい。

【0046】

【発明の効果】以上から明らかなように、通信を高速化し、処理オーバーヘッドとメモリ使用量を減らすことができる。

【図面の簡単な説明】

【図 1】TCP/IP プロトコルを使用し通信するホストのソフトウェア構成を示す図。

【図 2】高速通信を使用し通信するホストのソフトウェア構成を示す図。

【図 3】高速ソケットを使用し通信するホストのソフトウェア構成を示す図。

【図 4】MPI を使用し通信するホストのソフトウェア構成を示す図。

【図 5】MPI-Pro の通信方法を示す図。

【図 6】第一の問題を解決するための、通信方法切り替えしきい値のデータ長の転送を示す図。

【図 7】送信宛先を知らせるためのアドレス知らせメッセージとその応答を示す図。

【図 8】Winsock Direct の通信方法を示す図。

【図 9】Winsock Direct のプロトコルの詳細を示す図。

【図 10】Shah らによる方式の通信方法を示す図。

【図 11】本発明の通信方法を示す図。

【図 12】本発明の送信側の通信アルゴリズムを示す図。

【図 13】本発明の受信側の通信アルゴリズムを示す図。

【符号の説明】

10 : 送信ホスト

20 : 受信ホスト

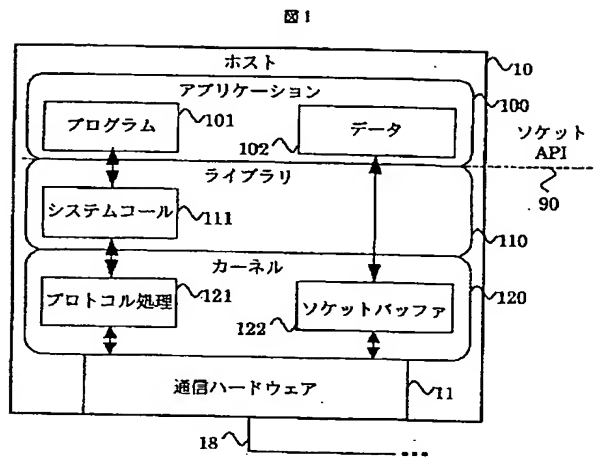
100, 103, 106, 200 : アプリケーション

120 : オペレーティング・システム・カーネル

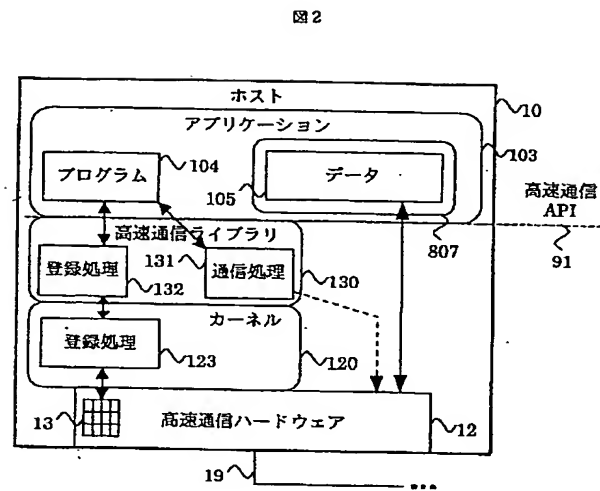
11 : 通信ハードウェア

12 : 高速通信ハードウェア。

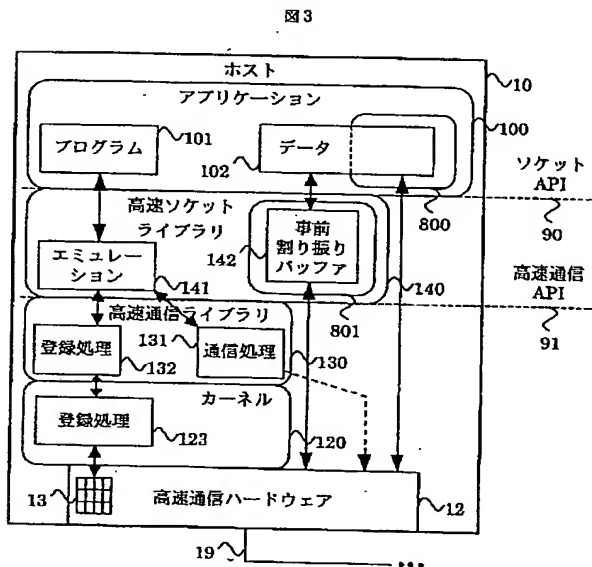
【図1】



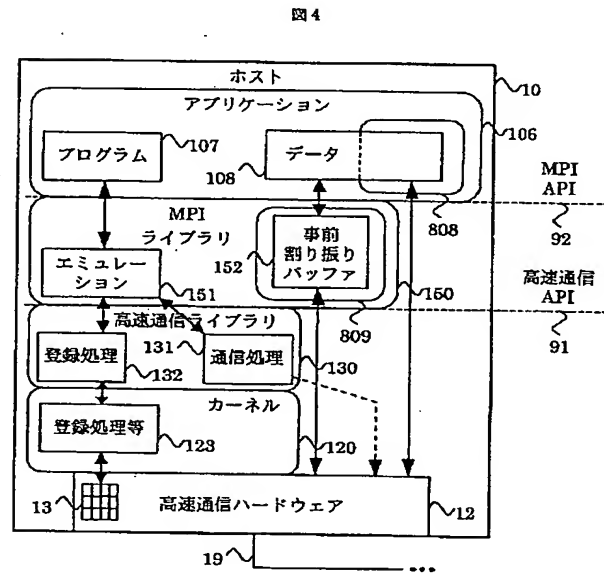
【図2】



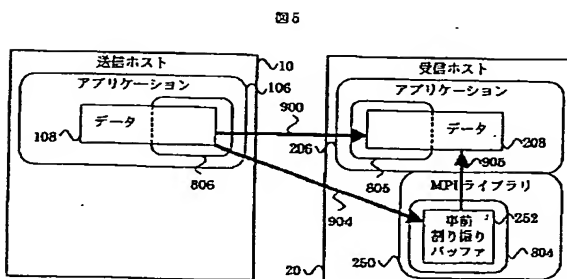
【図3】



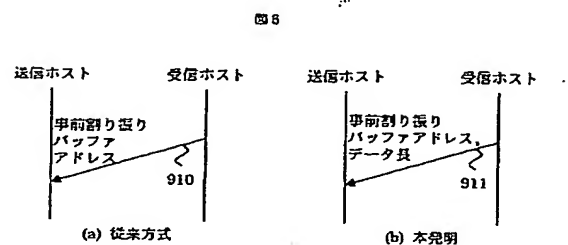
【図4】



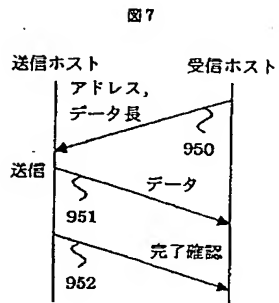
【図5】



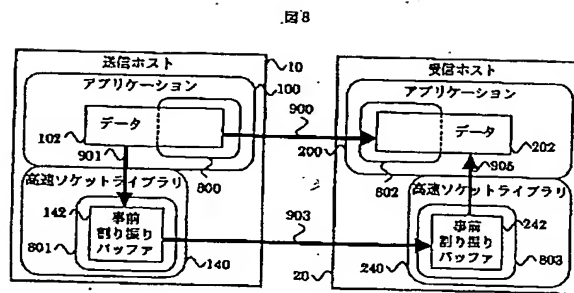
【図6】



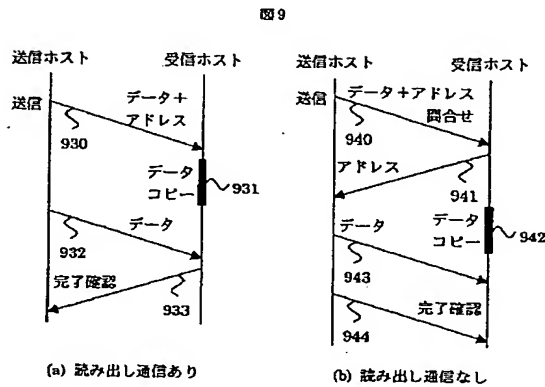
【図7】



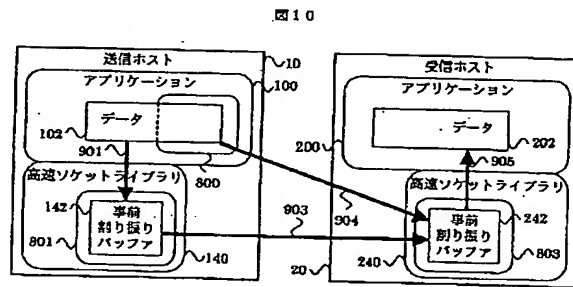
【図8】



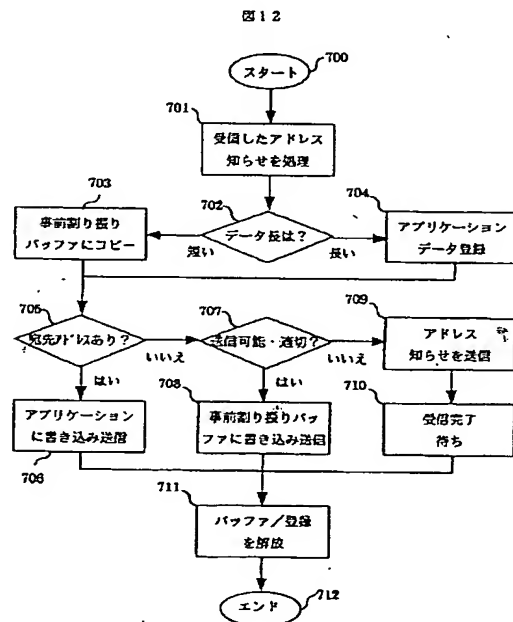
【図9】



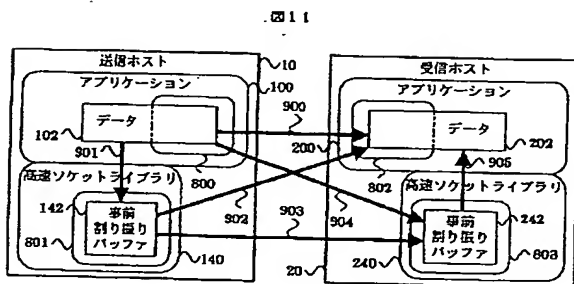
【図10】



【図12】



【図11】



【図13】

図13

